READS

Accelerator Real-Time Edge AI for Distributed Systems

Kyle Hazelwood (Mattson Thieme, Aakaash Narayanan)
AI for Accelerator Applications Workshop
January 14, 2022

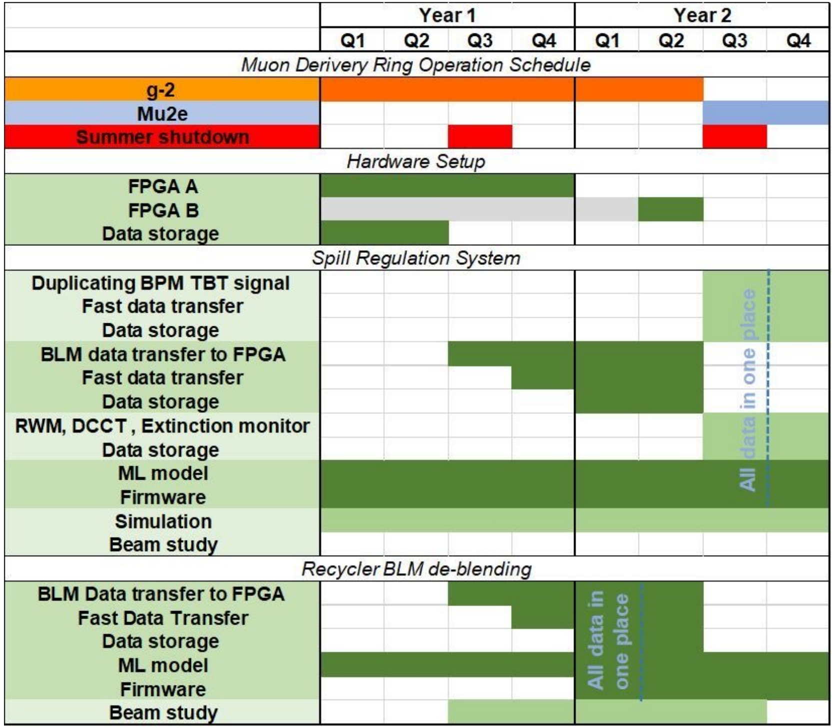
In partnership with:

NORTHWESTERN UNIVERSITY

READS Overview

- Funded by DOE 2020 FOA call for incoporating AI/ML into HEP accelerator facilities
- ~$1.5M over two years (we are in year 2)
- Two sub projects
  - Beam Loss Deblending for Main Injector and Recycler
  - Mu2e Spill Regulation
- Proposal https://arxiv.org/abs/2103.03928

This project aims to use machine learning implemented on fast hardware (FPGA) to provide real-time inferences using distributed readings from around the accelerator complex



Beam Loss Deblending for Main Injector and Recycler

- Main Injector and Recycler share an enclosure
- Both machines can and do often have high intensity beam in them simultaneous
- Both machines can generate significant beam loss
- The machine origin of a beam loss is often hard to distinguish
- Using time, location and state of the machine, machine experts can sometimes attribute loss to a particular machine
  - This suggests a Machine Learning (ML) model may be trainable to automatically attribute loss and replicate or improve upon the expert's ability
- Often losses from one machine end up tripping the machine permit of the other resulting in unnecessary beam downtime

The project aims to deploy a machine learning model on a FPGA that when fed streamed beam loss readings from around the Main Injector complex, will infer in real-time the machine loss origin

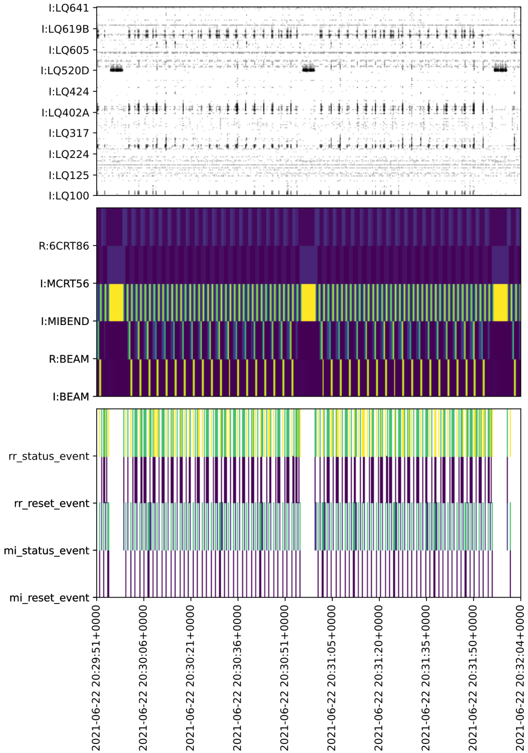
Main Injector tunnel
Recycler (top) Main Injector (bottom)

NORTHWESTERN UNIVERSITY Fermilab

Datasets

Data consists of TCLK (event), MDAT (machine readings), and 259 BLM readings

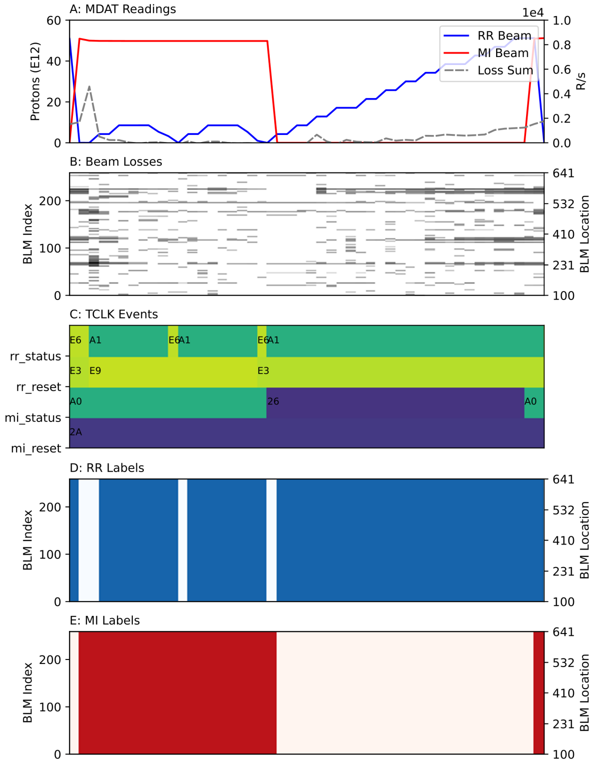
- Sample Dataset
  - 15 Hz
  - Data taken from machine operations
  - Continuously taken throughout the 2020/2021 run
- Study Dataset
  - 33 Hz
  - Data taken from 2021-06-22 dedicated end of run study
  - Timeline altered so that only Main Injector or Recycler had beam at any time
  - All beam loss attributable to a machine
  - Beam losses purposefully generated in both machines using various machine misconfigurations to not bias a model towards standard running



Few minute example of studies data

Datasets (continued)

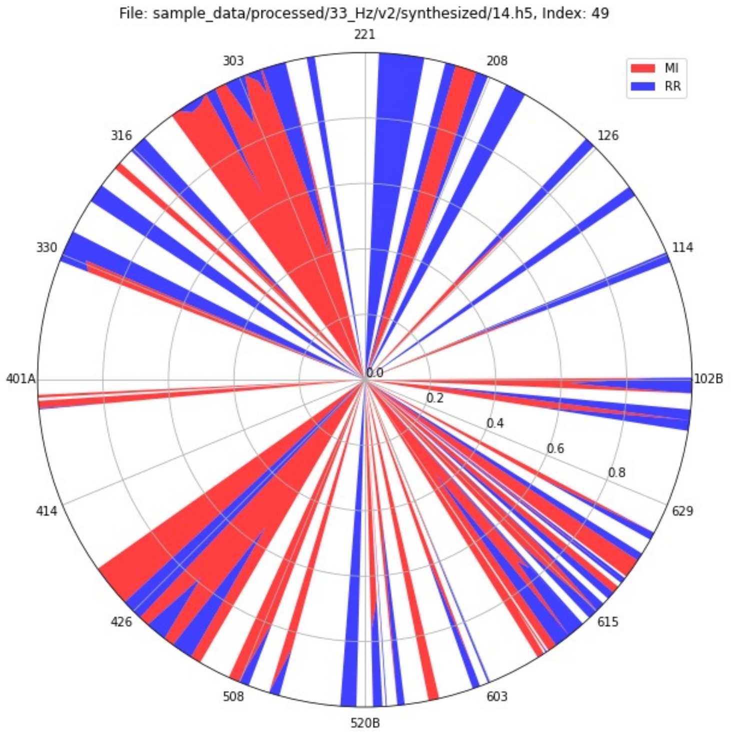
- Data labeling done using multi-threaded data processing code
- Labeling uses beam intensity, other MDAT readings and TCLK event thresholds to determine whether loss was possible from a machine
- Outputs a fraction label for each BLM, per machine, per data time sample
  - 0.0 for loss that ***did not*** come from machine
  - 1.0 for loss that ***could*** have come from machine
  - Times for which data processor outputs NaN for both machines are referred to as “unknown”
  - Unknown data is not used for training or validation but rather for testing model inference (These are the times we can’t accurately attribute now)



Example of data labeling

Datasets (continued)

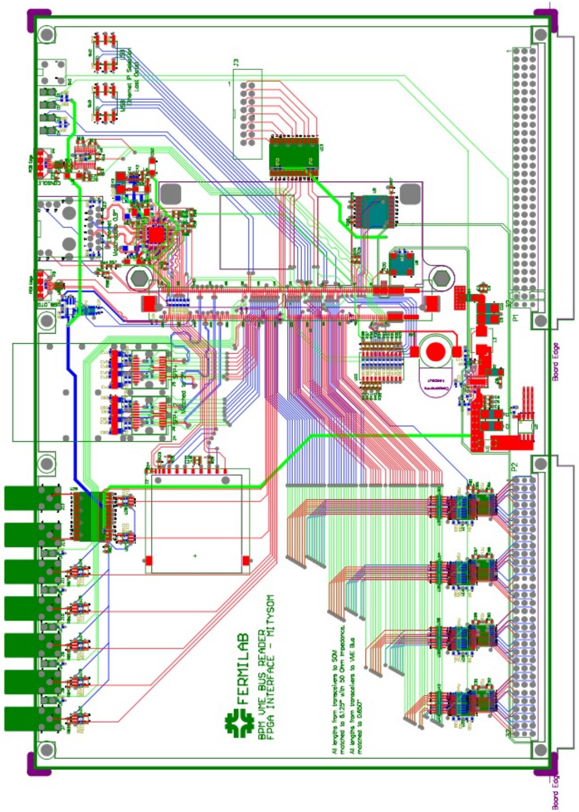
- Synthesized Dataset
  - 33 Hz I 333 Hz
  - Data most like Final dataset
  - Using Sample, Study and (eventually) Final Datasets
  - Use known losses (attributed to one machine) and sum with known losses attributable to the other machine
  - Resulting labels are percentages of loss per BLM attributed per machine
  - Will be used to perform semi-supervised model training and will supplemer operations data



Example of synthesized data labeling

VME Bus Reader (Pirate) Card

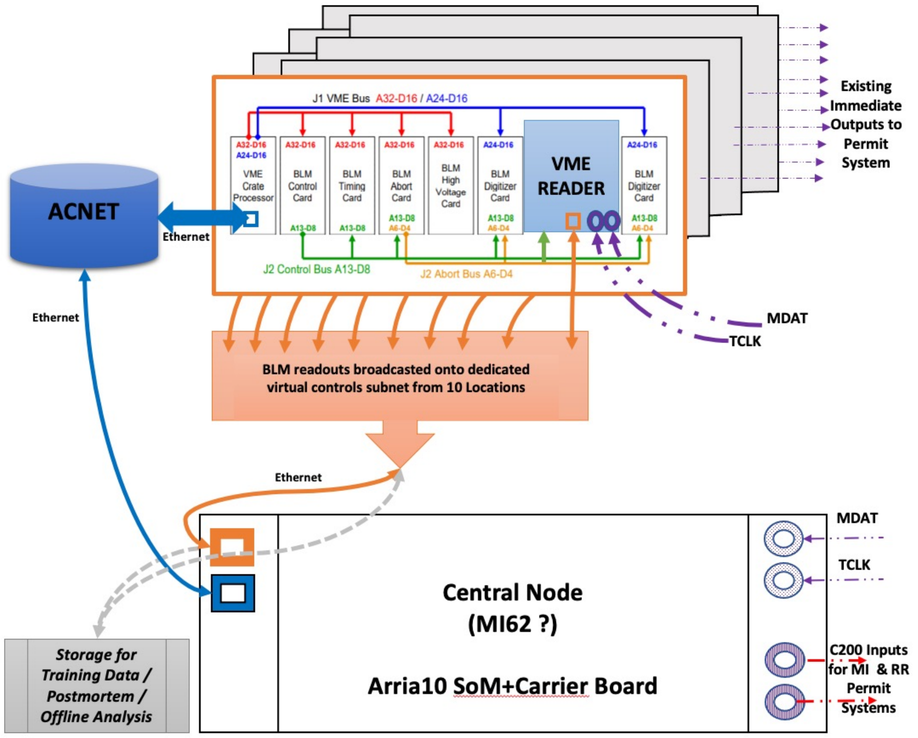
- Existing BLM nodes can’t handle the data IO
  - It was beyond the scope of this project to modify the BLM nodes
- 333 Hz (BLM node digitizer poll rate)
- Streams to disk for training and to eventual central FPGA node for inference (< 3mS latency)
- Card are fabricated
- Finishing up firmware and testing
- Should be implemented ring wide by Spring



VME Bus Reader (Pirate) card

Central Node

- Central node is an Aria10 FPGA SOM
- Board has an HPS and FPGA
- ML model will be deployed on FPGA
- Two ethernet ports
  - One dedicated to ACNET, connected to HPS
  - One dedicated to Pirate Card stream, direct to FPGA fabric
- Has inputs for MDAT and TCLK
- Has TTL outputs intended for MI and RR c200 permit input
- Node will broadcast inferences at 333Hz in both DDCP protocol (for future training and validation data) as well as ACNET readings

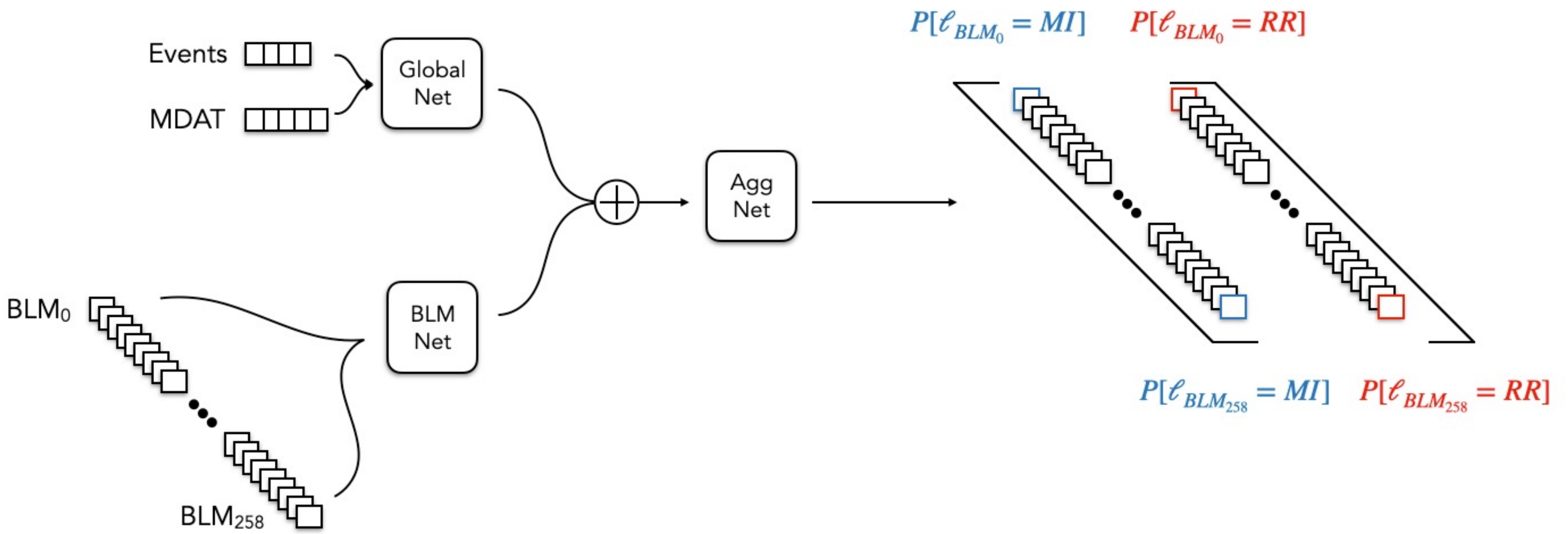


Central node data paths

NORTHWESTERN UNIVERSITY Fermilab

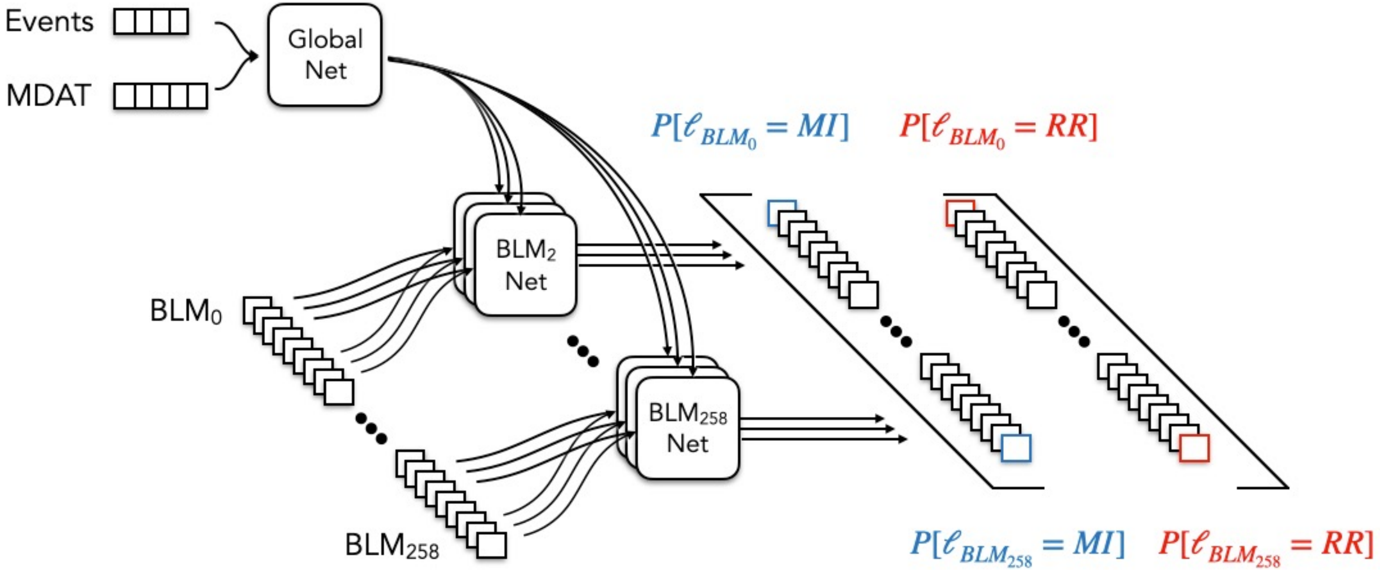
ML Model Architecture: Phase 1, Data-Type-Specific Aggregation

Objective: Assign BLM-wise probabilities for that loss originating in MI/RR



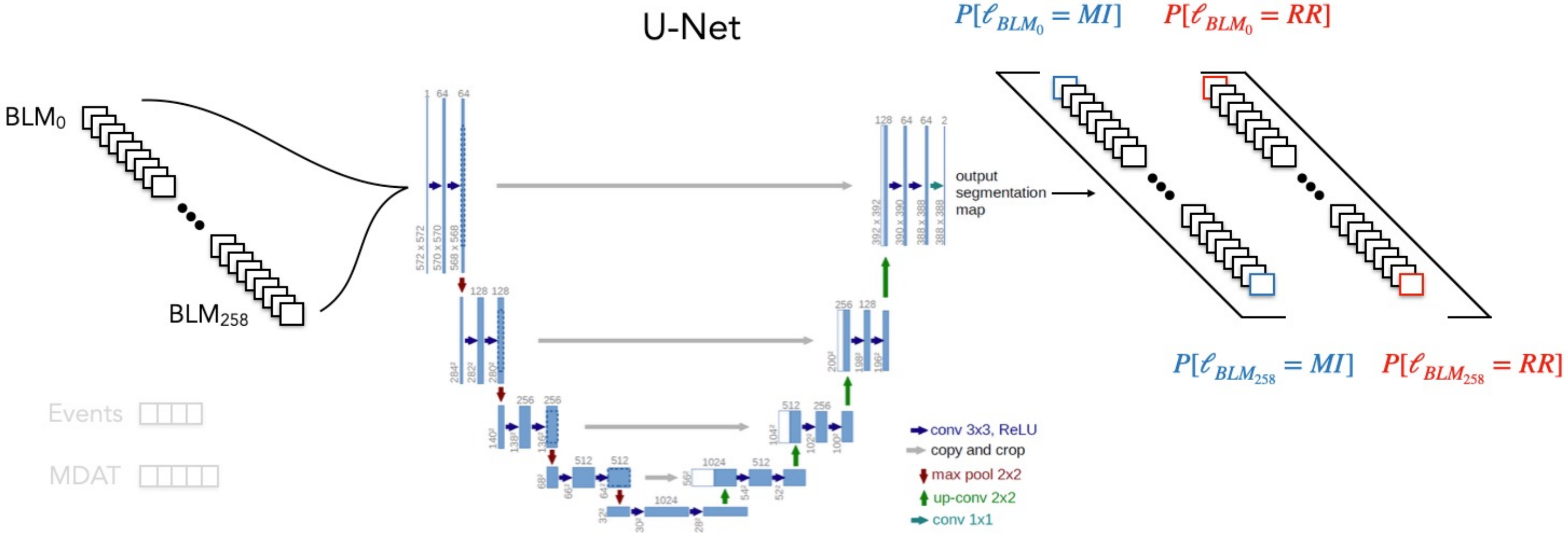
ML Model Architecture: Phase 2, Forcing Locality

Objective: Assign BLM-wise probabilities for that loss originating in MI/RR



ML Model Architecture: Phase 3, Varying Receptive Fields

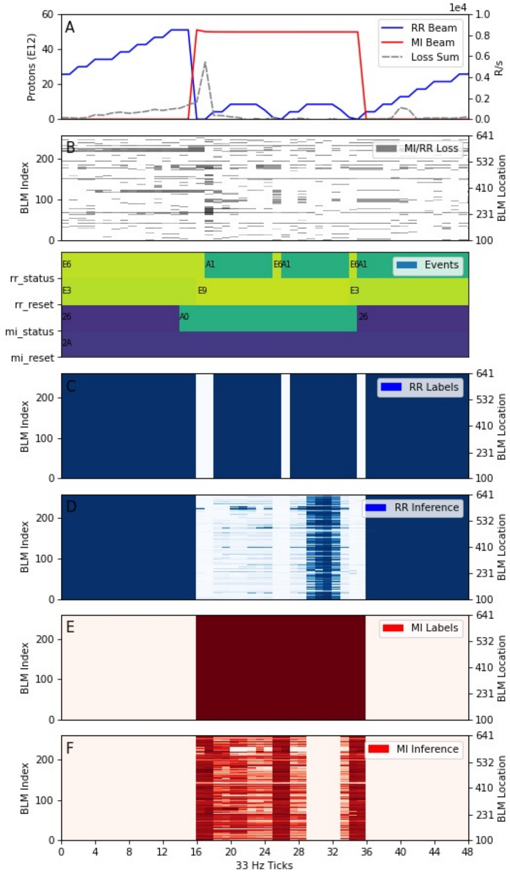
Objective: Assign BLM-wise probabilities for that loss originating in MI/RR



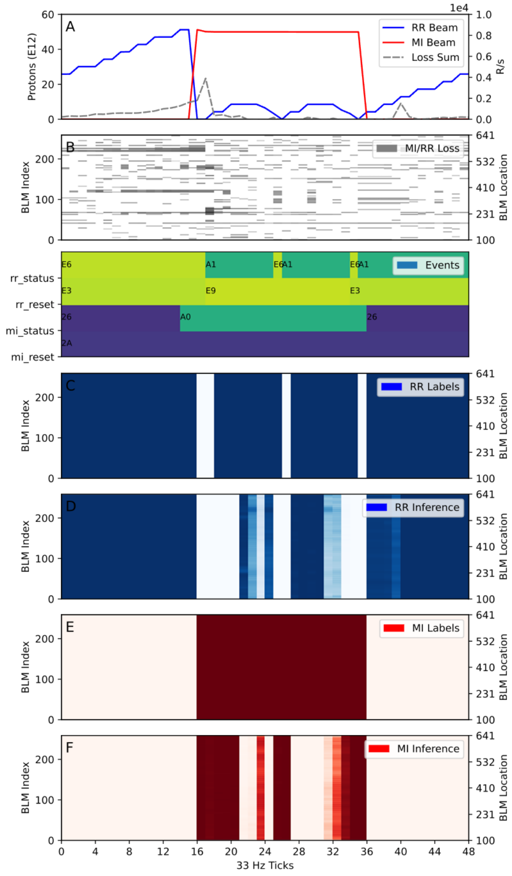
NORTHWESTERN UNIVERSITY Fermilab

ML Model Inference

- Preliminary models are promising
- Phase 1 (CNN) models recognized state transitions well, but inference was very homogenous across BLMs, often attributed losses only to one machine or the other
- Phase 2 (ManyModels) models also recognized state transitions well and picked up on local BLM patterns but lost all global loss pattern context
- Phase 3 (UNet) recognizes both local and global BLM patterns well and correctly picks up on state transitions despite being only trained on BLM data (no state data)!

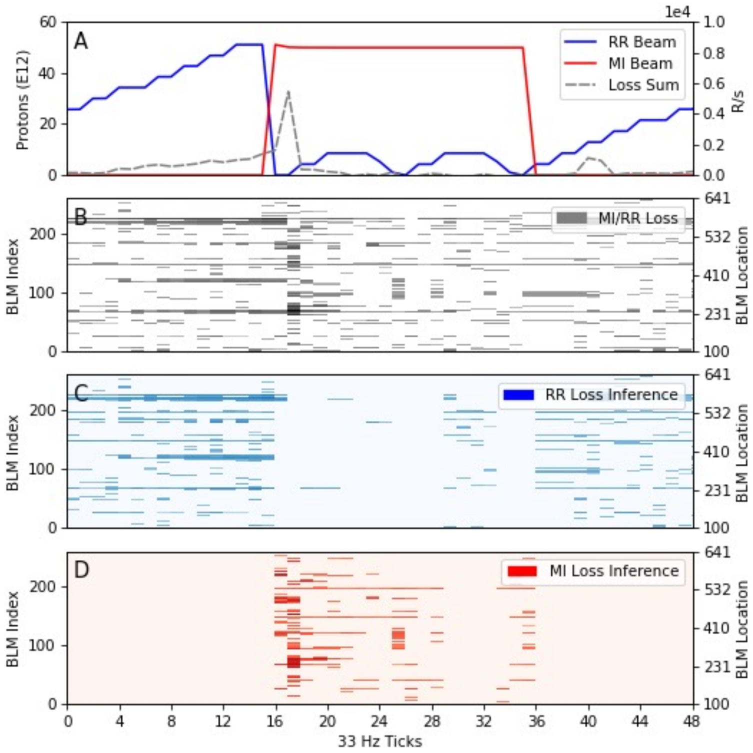


Phase 2 (ManyModels) inference



Phase 3 (UNet) inference

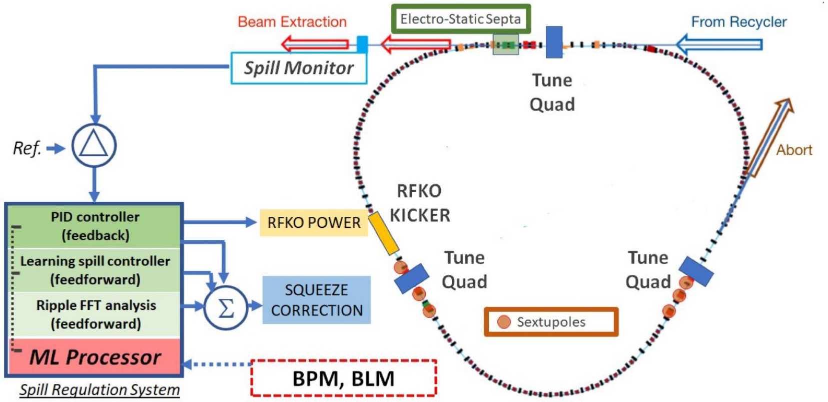
ML Model Inference (continued)



Example model inferred losses

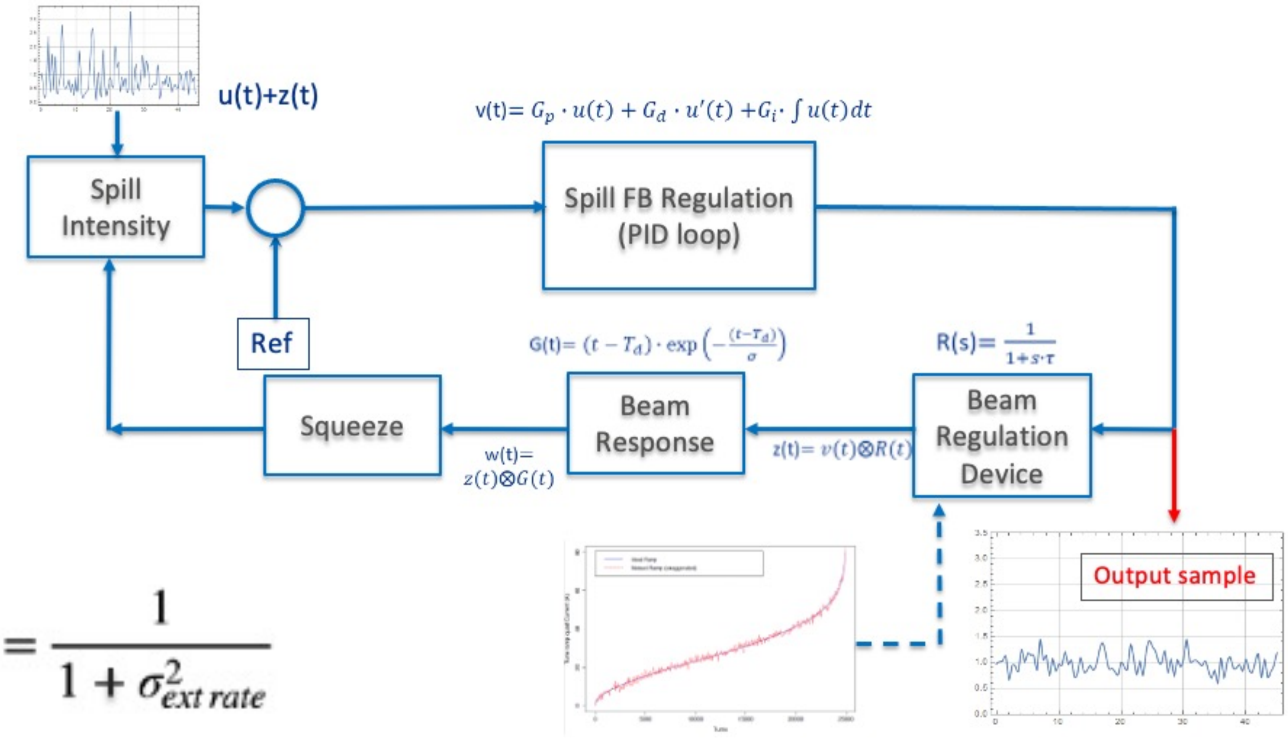
Mu2e Slow Spill Regulation

- Resonant beam extraction from Delivery Ring to the Mu2e experiment
- Very fast spill, 43mS
- 8kW beam power
- Very tight requirements on spill quality to avoid negative impacts
  - Detector pile-up
  - Reconstruction inefficiency
  - Dead time



The project aims to deploy a ML regulation system that optimizes or improves upon traditional PID loop controllers at correcting for higher frequency noise in the spill and raises the Spill Duty Factor (SDF)

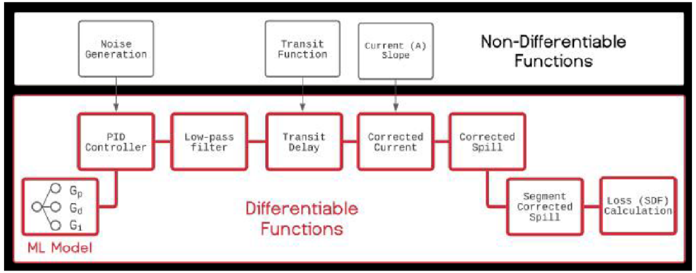
Spill Regulation Model (Spill Simulator)



$$SDF = \frac{1}{1 + \sigma^2_{ext\ rate}}$$

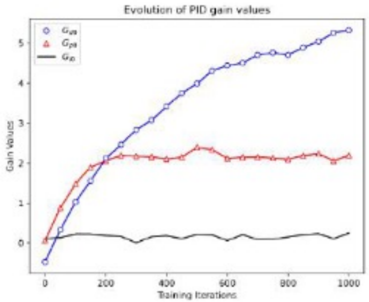
ML Model Architecture: Phase 1, PID Gains Optimization

Objective: Optimize PID gains using ML with differentiable spill simulator

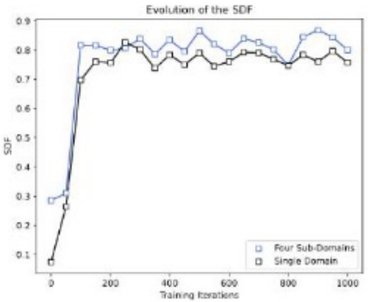


Final **SDF=74%** with a single domain
Final **SDF=83%** with 4 subdomains

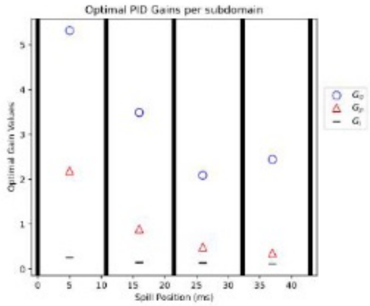
$$\ell = (1 - SDF)^2$$



Evolution of gains in first subdomain.



Evolution of SDF of full spill.



Optimized gains in all 4 subdomains at end of last iteration.

NORTHWESTERN UNIVERSITY

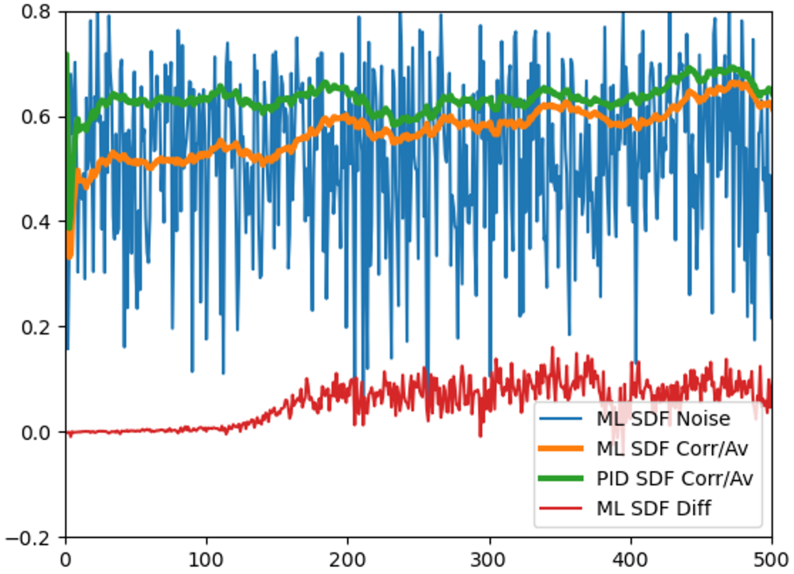
Fermilab

ML Model Architecture: Phase 2, ML Defined Regulation

Objective: Replace PID controller with Supervised Learning ML process

- Ablation studies
  - Architectures (MLP, CNN, RNN, etc)
  - Input parameters (slopes, relative position, etc)
  - Window Sizes, model depth, etc
  - Optimizers, LR schedulers, etc
  - Many more

ML model shows similar performance as the PID loop



500 Spills

ML Model Architecture: Phase 3, Investigate Reinforcement Learning

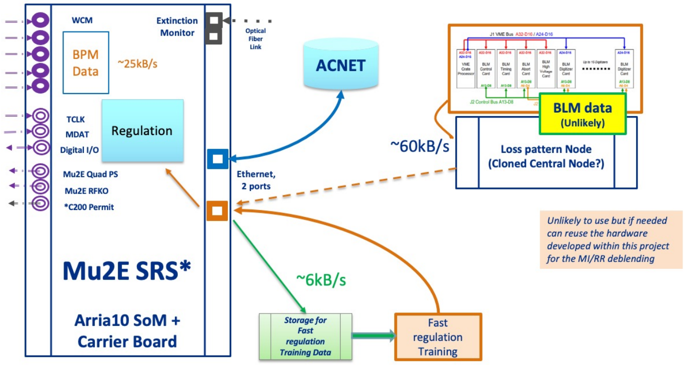
Objective: Switch from Supervised Learning (SL) to Reinforcement Learning (RL) ML scheme

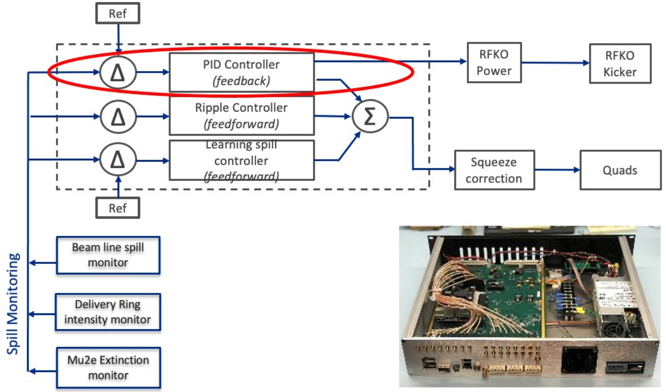
- Has potential to find more optimal policies
- Could learn from future real data
- Allows spill simulator to be non-differentiable (faster/simpler)

In progress…

Regulation Node

- Aria10 SOC/SOM board (Same as deblending central node)
- 3 independent controllers
- Two beam control systems
- Optical input for Spill Monitor signal
- Up to 10kHz BW open loop
- ML agent
- Board SW in progress (covered separately)





Summary

- Beam Loss Deblending for Main Injector and Recycler
  - A robust dataset collection and processing scheme has been developed and is in use to collect and label operations data
  - High frequency data is expected soon with the deployment of our BLM VME Bus reader (Pirate) cards
  - Various model architectures have been investigated with UNet emerging as the possible final design
  - Preliminary trained models show great promise
  - Progress thus far was presented at IPAC'21 (paper MOPAB288) and another paper is in the works for Spring 2022
  - We are on schedule to commission a final ML model deployed on a central node Summer or Fall 2022

- Mu2e Spill Regulation
  - Using differentiable spill simulator
  - PID optimization done, simulated SDF 80+%
  - Direct Supervised Learning (SL) ML confroller is comparable to more traditional PID loop controller
  - Investigating Reinforcement Learning ML to improve upon performance of PID and SL ML controllers
  - Progress thus far was presented at IPAC'21 (paper THPAB243)

The READS Team

Fermilab
Aakaash Narayanan (NIU)
Kyle Hazelwood
Aisha Ibrahim
Vladimir Nagaslaev
Dennis Nicklaus
Peter Prieto
Kiyomi Seiya
Randy Thurman-Keup
Nhan Tran
Brian Schupbach
Pooja Swamy
Jose Berlioz
Mark Austin
Tia Micelli
Bill Pellico
Andrea Saewart

Northwestern University
Han Liu
Seda Memik
Mattson Thieme
Rui Shi